SAFA: Sample-Adaptive Feature Augmentation for Long-Tailed Image Classification

Yan Hong[1], Jianfu Zhang[⋆1], Zhongyi Sun[2], and Ke Yan[⋆2]

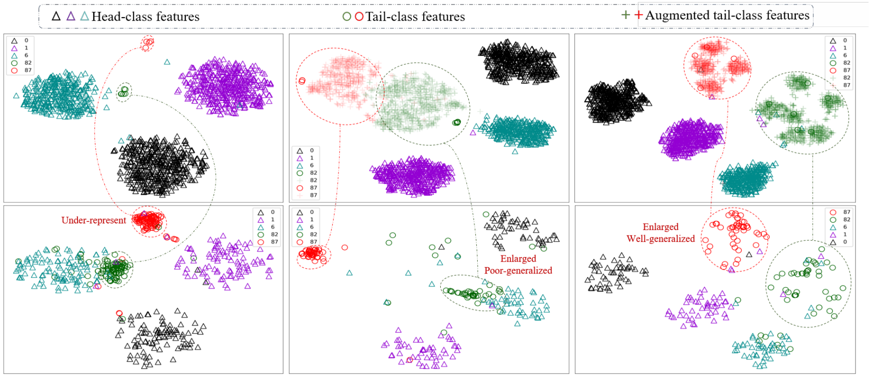
[1] MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, China
yanhong.sjtu@gmail.com, c.sis@sjtu.edu.cn
[2] Tencent Youtu Lab, China
{zhongyisun, kerwinyan}@tencent.com

Abstract. Imbalanced datasets with long-tailed distribution widely exist in practice, posing great challenges for deep networks on how to handle the biased predictions between head (majority, frequent) classes and tail (minority, rare) classes. Feature space of tail classes learned by deep networks is usually under-represented, causing heterogeneous performance among different classes. Existing methods augment tail-class features to compensate tail classes on feature space, but these methods fail to generalize on test phase. To mitigate this problem, we propose a novel Sample-Adaptive Feature Augmentation (SAFA) to augment features for tail classes resulting in ameliorating the classifier performance. SAFA aims to extract diverse and transferable semantic directions from head classes, and adaptively translate tail-class features along extracted semantic directions for augmentation. SAFA leverages a recycling training scheme ensuring augmented features are sample-specific. Contrastive loss ensures the transferable semantic directions are class-irrelevant and mode seeking loss is adopted to produce diverse tail-class features and enlarge the feature space of tail classes. The proposed SAFA as a plug-in is convenient and versatile to be combined with different methods during training phase without additional computational burden at test time. By leveraging SAFA, we obtain outstanding results on CIFAR-LT-10, CIFAR-LT-100, Places-LT, ImageNet-LT, and iNaturalist2018.

1 Introduction

With the development of deep convolutional neural networks (CNNs) [16] trained with large-scale datasets [32], computer vision research has been propelled forward significantly in recent years. These large-scale datasets are usually well-designed with the number of instances in each class balanced artificially, which however is inconsistent with the real-world scenarios. It is common that the images of some categories are easy to be collected while some others are difficult, resulting in the number of samples in each head class being far greater than the number of samples in each tail class, as shown in Fig. 1 (a). Due to the insufficient information of tail classes, CNNs' feature space for tail classes is

[⋆] Corresponding authors.



(a) w/o Augmentation (b) w/o Sample-Specific (c) Sample-Specific

Fig. 1. Motivation of this work: We select three head classes and two tail classes from CIFAR-LT-100 dataset [25] and plot t-SNE [28] visualization to compare methods including: (a) LDAM [6], reweighting-based method without tail-class augmentation in training phase; (b) RSG [38], augmentation-based method without sample-specific augmentation; and (c) SAFA, our proposed augmentation-based method with sample-specific augmentation. (a) *"w/o Augmentation"*: Imbalanced distributions for head-class samples and tail-class samples cause *CNNs under-represent tail classes in feature space.* (b) *"w/o Sample-Specific"*: CNNs enlarge feature spaces for tail classes with augmented tail-class features. However, these augmented tail-class features are not sample-specific and distracting from real tail-class features, making *the feature space for tail classes fail to generalize to test phase and is still under-represented.* (c) *"Sample-Specific"*: Sample-specific augmented features recover the distribution of limited tail-class samples, which *enlarges feature space of tail classes and generalizes better to test phase*, helping CNNs to perform more homogeneously across different classes

under-represented and the decision boundary is biased to head classes, leading to poor classification performance on tail classes.

To address the issue of imbalance data distribution, a natural solution is augmenting training samples to compensate tail classes in feature space. Data augmentation techniques like cropping, mirroring and mixup [16,18,44] are adopted to alleviate data imbalance problem. However, these conventional data augmentation techniques are typically performed inside each tail class without considering information in head classes. As a result, the diversity of augmented samples is inherently limited by the insufficient training samples in tail classes so that the augmented data can not recover the data distribution of tail classes. Considering that head class with amounts of samples providing diverse intra-class variance, previous works [23,9,10,33,39,43,38] adopt different methods to enlarge feature space of tail classes by generating new features for tail classes during training via transferring intra-class variance information from head classes to tail classes. [33,43] utilize feature variation information, such as different poses or lighting conditions, among samples from the same head class to generate new tail-class

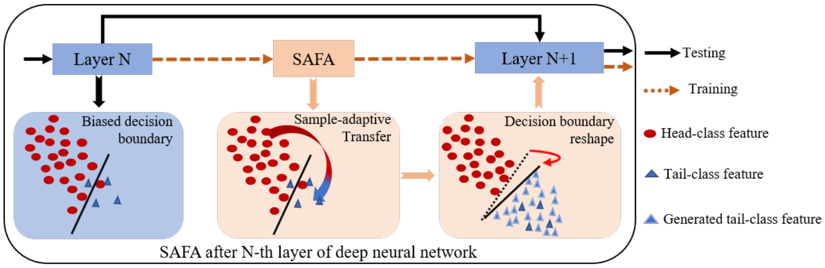


Fig. 2. The illustration of integrating our proposed SAFA into the N-th layer in deep network to produce diverse and effective tail-class features to reshape the feature space. Our SAFA is only used during training denoted by orange dot line, leaving no computational burden at test time denoted as black solid line

features. However, these methods did not introduce any mechanisms to ensure the variation information obtained from head classes is *class-irrelevant*. The augmented tail-class samples may shift to other classes due to the class-relevant information from head classes, hurting the performance of CNN classifiers. Also, these approaches are not in an end-to-end manner. To augment tail-classes with class-irrelevant information, noise vectors are used in [39] to encode the sample variation information. But noise vectors are too random to reflect the true variations among images, using such noise vectors for generation can possibly generate unstable or low-quality features. In [38], a feature augmentation module is integrated into CNNs for end-to-end training, the variation information extracted by removing the centers of each class and a vector transformation module is used to enlarge the distance between feature variance and tail-class features. All of abovementioned methods adopt a direct combination between the intra-class variance extracted from head class and random tail-class samples to produce abundant augmented features belonging to tail classes. Whereas, these augmented features are *not sample-specific*: the incompatibility between the tail-class sample with applied intra-class variance causes implausible augmented features distracting from real features in feature space, as shown Fig. 1 (b). *CNNs do enlarge (resp., reduce) feature space of tail classes (resp., head classes) during training phase with these non-sample-specific augmented features, but unfortunately fail to generalize the feature space on test phase that the tail classes are still under-represented.*

In this paper, to alleviate these limitations, we propose a novel semantic Sample-Adaptive Feature Augmentation (SAFA) to generate reliable and diverse augmented features for tail classes during training phase to enlarge the under-represented feature space of tail classes and improve classifiers with less biased decision boundary. SAFA is a novel plug-in approach, which is convenient to be integrated into various networks to effectively augment tail classes without additional computational burden in testing phase, as shown in Fig. 2. Note that we only show a simple CNN in Fig. 2, but SAFA can be used in any network architecture. SAFA aims to extract diverse and transferable semantic directions (*e.g.,* intra-class transformation) from head-class features and translate tail-class samples along extracted directions adaptively to produce diverse and effective features. SAFA is formulated by auto-encoder structure consisting of an sample-specific encoder and a sample-adaptive generator. The encoder is used to extract transferable class-irrelevant information from head classes, while the sample-adaptive generator is designed to correct extracted variance information to produce sample-specific features of tail classes. SAFA leverages a recycling training scheme enforcing consistency of the relevant semantics before and after translation and ensuring augmented features are sample-specific. Contrastive loss ensures the transferable semantic directions are class-irrelevant and mode seeking loss is adopted to exploit diverse semantic directions, producing diverse tail-class features and enlarging the feature space of tail classes. In Fig. 1 (c), we demonstrate the effect of SAFA. *SAFA is able to generate diverse and effective augmented features and recover the real distribution of tail classes, enlarging the feature space of tail classes and generalizing promisingly in test phase.* The proposed SAFA as a plug-in is convenient and versatile to be combined with different architectures and loss functions during training phase without additional computational burden at test time. With extensive experimental evaluations, we verify the effectiveness of SAFA: SAFA obtains outstanding results on Imbalanced CIFAR, Places-LT, ImageNet-LT, and iNaturalist2018.

2 Related Work

2.1 Long-tail Classification Methods

Re-sampling Over-sampling the tail classes [34,4,5] or under-sampling the head classes [15,21,4] strategies are widely used to balance the data distribution for imbalanced datasets. Although being effective, over-sampling might result in over-fitting of tail classes while under-sampling may weaken the feature learning of head classes due to the absence of valuable samples [42,6,7,11].

Re-weighting Reweighting-based methods aim to assign weights to training samples on either class or sample level. A classic scheme is to reweight the classes with the weights that are inversely proportional to their frequencies [17,40]. The method in [11] further improves this scheme with proposed effective number. L2RW [31] is designed to assign weights to examples sample-wisely based on the gradient directions. Meta-class-weight [20] exploits meta-learning to estimate precise class-wise weights, while [6] allocate large margins to tail classes. Apart from above works, Focal Loss [26] and meta-weight-net [35] assign weights to examples sample-wisely. In addition, for learning better representations, some approaches propose to separate the training into two stages: representation learning and classifier re-balancing learning [6,20,12,22]. BBN [48] further unifies the two stages to form a cumulative learning strategy.

Augmentation Data augmentation is widely adopted to CNNs for alleviating over-fitting. For example, rotation and horizontal flipping are employed for maintaining the prediction invariant of CNNs [16,18,36]. In complementary to the traditional data augmentation, semantic data augmentation that performs semantic altering is also effective for enhancing classifier performance[2,41]. A hallucinator [39] was designed to generate new samples for tail classes. It uses samples from tail classes and noise vectors to produce new hallucinated samples for tail classes. A Delta-encoder framework [33] was proposed for generating new samples. It is first trained to reconstruct the pre-computed feature vector of input images from head classes. Thereafter, it is used to generate new samples by combining the tail-class samples, and the newly generated ones are further used to train the classifier. A feature transfer learning (FTL) framework [43] was proposed to transfer the intra-class variance from head classes to tail classes by generating new tail-class samples. Our methods can be categorized as augmentation-based methods, which mainly focus on augmenting tail-class samples to overcome imbalance issue. Different from other augmentation methods that simply apply the same transformation (*e.g.,* adding random noises) to all tail-class samples, we distinguish different samples and design a sample-adaptive augmentation method to produce effective and diverse augmented tail-class samples. Our method fully considers individual differences combined with intra-class variance to generate semantically rational augmentations.

2.2 Semantic Transformations in Deep Feature Space

Our work is motivated by the fact that high-level representations learned by deep convolutional networks can potentially capture abstractions with semantics [3]. In fact, translating deep features along certain directions is shown to be corresponding to performing meaningful semantic transformations on the input images. For example, deep feature interpolation [8,49] leverages simple interpolations of deep features from pre-trained neural networks to achieve semantic image transformations. Variational Auto-Encoder (VAE) [24] and Generative Adversarial Network (GAN) based methods [14] establish a latent representation corresponding to the abstractions of images, which can be manipulated to edit the semantics of images. Generally, these methods reveal that certain directions in the deep feature space correspond to meaningful semantic transformations, and can be leveraged to perform semantic data augmentation. In this work, we focus on learn adaptive semantic transformations for tail-class by leveraging diverse class-invariant features from head classes.

3 Methodology

Given an imbalanced training dataset $\mathbb{S} = \{\boldsymbol{x}^i, y^i|_{i=1}^n\}$, where $y^i \in \{1, \cdots, C\}$ is the label of i-th sample $\boldsymbol{x}^i$, where C is the number of classes, and n_c denotes the number of samples belongs to the c-th class. We assume that the classes are sorted by cardinality in a decreasing order, *i.e.,* $n_{i+1} \leq n_i$. The data obeys the long tail distribution, *i.e.,* most samples belong to only a few head classes denoted as $\{\boldsymbol{x}_h^i\}$ and data of the other tail classes represented as $\{\boldsymbol{x}_t^i\}$ only has a few samples. Feeding head-class samples $\{\boldsymbol{x}_h^i\}$ (*resp.,* tail-class samples $\{\boldsymbol{x}_t^i\}$)

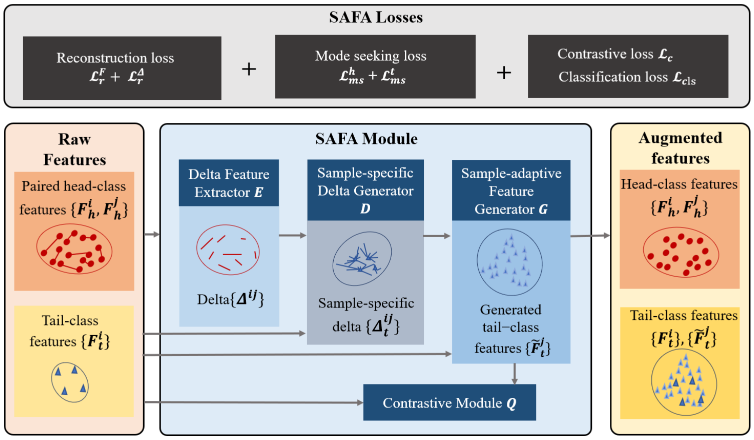

SAFA Losses
Reconstruction loss $\mathcal{L}_r^F + \mathcal{L}_r^\Delta$
+
Mode seeking loss $\mathcal{L}_{ms}^h + \mathcal{L}_{ms}^t$
+
Contrastive loss $\mathcal{L}_c$
Classification loss $\mathcal{L}_{cls}$
Raw Features
Paired head-class features $\{F_h^i, F_h^j\}$
Tail-class features $\{F_t^i\}$
SAFA Module
Delta Feature Extractor E
Delta$\{\Delta^{ij}\}$
Sample-specific Delta Generator D
Sample-specific delta $\{\Delta_t^{ij}\}$
Sample-adaptive Feature Generator G
Generated tail−class features $\{\tilde{F}_t^j\}$
Contrastive Module Q
Augmented features
Head-class features $\{F_h^i, F_h^j\}$
Tail-class features $\{F_t^i\}, \{\tilde{F}_t^j\}$

Fig. 3. The framework of SAFA, including a delta extraction module E, a sample-specific delta generator D, and a sample-adaptive generator G, and a contrastive module Q. E is used to extract *class-irrelevant delta* $\boldsymbol{\Delta}^{ij}$ from head-class pairs $\{\boldsymbol{F}_h^i, \boldsymbol{F}_h^j\}$, D is applied to combined extracted $\boldsymbol{\Delta}^{ij}$ with tail-class feature $\boldsymbol{F}_t^i$ to produce *sample-specific delta* $\boldsymbol{\Delta}_t^{ij}$, which coupled with $\boldsymbol{F}_t^i$ is fed into sample-adaptive generator G to generated *sample-specific tail-class feature* $\tilde{\boldsymbol{F}}_t^j$

into CNNs, the corresponding feature maps from specific layer of backbone are denoted as $\{\boldsymbol{F}_h^i\}$ (*resp.*, $\{\boldsymbol{F}_t^i\}$).

3.1 SAFA: Sample-Specific Feature Augmentation

In this section, we introduce how to integrate SAFA into CNNs for producing diverse tail-class features to effectively enlarge tail-class feature space during training phase and generalize to test phase. Our SAFA is inspired by [33], in which intra-class transformation (*i.e.*, the difference between two samples within the same category) is called "delta". Deltas are extracted from paired samples of the same class, in which delta is the additional information required to reconstruct one sample of the pairs from another sample. In [33], deltas are directly combined with random target-class samples to generate new features for target classes. However, the effect of delta may depend on the combined target-class samples [1], that is, an effective delta for one sample may be unsuitable for another sample. On one hand, the extracted deltas are different in semantic scale, *e.g.,* different degrees (90 or 180) pose rotation. On the other hand, the difficulties of translating samples from tail classes with different-scale semantic directions are different, *e.g.,* translating a dog with left face to a dog with left face may be easier than another dog with frontal face. Naive augmentation may lead to corrupted features or features without class-preserving characteristic, which are distracting from real tail-class features, as shown in Fig. 1 (b).

To extract effective and transferable deltas, and adaptively apply these deltas to tail-class samples to produce effective augmented features, we propose SAFA as illustrated in Fig. 3. SAFA consists of a delta feature extractor E, a sample-specific delta generator D, a sample-adaptive feature generator G and a contrastive module Q. All these modules are built up with Conv-BN-ReLU-Conv layers (Q has an additional FC layer). During training, given a random pair of feature maps $\{\boldsymbol{F}_h^i, \boldsymbol{F}_h^j\}$ from the same head class (*i.e.*, $y_h^i = y_h^j$), the delta extraction module E is used to extract *class-irrelevant delta* $\boldsymbol{\Delta}^{ij}$, which combined with random tail-class feature maps $\boldsymbol{F}_t^i$ fed into the sample-specific delta generator D to generate *sample-specific delta* $\boldsymbol{\Delta}_t^{ij}$. After that, $\boldsymbol{\Delta}_t^{ij}$ combined with $\boldsymbol{F}_t^i$ for the sample-adaptive generation module G to produce *sample-specific tail-class features* $\tilde{\boldsymbol{F}}_t^j$. Finally, real feature maps $\boldsymbol{F}$ are coupled with augmented tail-class feature maps $\tilde{\boldsymbol{F}}_t$ are fed into deeper layers of the network.

To ensure the transferability of extracted delta, a modified recycle reconstruction loss [19] is adopted to ensure that delta encoder and sample-adaptive generator are inverses of each other. As shown in Fig. 4, extracted delta $\boldsymbol{\Delta}^{ij}$ from $\{\boldsymbol{F}_h^i, \boldsymbol{F}_h^j\}$ are reconstructed from fake tail-class pair $\{\tilde{\boldsymbol{F}}_t^j, \boldsymbol{F}_t^j\}$ as $\hat{\boldsymbol{\Delta}}^{ij}$. Further, $\hat{\boldsymbol{\Delta}}^{ij}$ is combined with $\boldsymbol{F}_h^i$ to reconstruct $\boldsymbol{F}_h^j$ by $\hat{\boldsymbol{F}}_h^j$. In this way, delta information and sample information are reconstructed bidirectionally, effectively improving the transferability of extracted delta information and enforcing the generated features to be sample-specific. To ensure the class-preserving characteristic of augmented tail-class samples, we introduce contrastive learning in Q to push away paired samples from different classes while pairs from the same class are dragged in. To further improve the diversity of augmented samples and enlarge the feature space of tail classes, a modified mode seeking loss [29] is integrated into SAFA by maximizing the ratio of the distance between augmented tail-class samples with respect to the distance between extracted deltas.

The overall objective function of SAFA can be given as follows,

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_{\text{r}} + \lambda_2 \mathcal{L}_{\text{ms}}^t + \lambda_3 \mathcal{L}_{\text{ms}}^h + \lambda_4 \mathcal{L}_c \tag{1}$$

where $\mathcal{L}_{cls}$ denotes any classification loss, such as softmax with cross-entropy loss, focal loss [26], LDAM [6]; λ_1 (*resp.,* λ_2, λ_3, λ_4) denote coefficient; $\mathcal{L}_{\text{r}}$, $\mathcal{L}_{\text{ms}}$ and $\mathcal{L}_c$ are cycle reconstruction loss, mode seeking loss and contrastive loss respectively, which will be introduced in the next subsection.

3.2 Module Details and Objective Functions

Class-irrelevant delta extraction The delta feature extraction module E aims to capture diverse and transferable delta information. Given a pair of feature $\{\boldsymbol{F}_h^i, \boldsymbol{F}_h^j\} \in \mathcal{R}^{C\times W\times H}$ from the same head class, where C (*resp.,* W, H) denotes the channel (*resp.,* width, height) dimension. The delta feature extraction module E is used to extract delta feature $\boldsymbol{\Delta}^{ij}$:

$$\boldsymbol{\Delta}^{ij} = E(\boldsymbol{F}_h^i - \boldsymbol{F}_h^j), \tag{2}$$

where $\boldsymbol{\Delta}^{ij} \in \mathcal{R}^{C_{\boldsymbol{\Delta}}\times W\times H}$, and $C_{\boldsymbol{\Delta}}$ represents the dimension of delta features. Such extracted delta feature $\boldsymbol{\Delta}^{ij}$ captures the variance (*i.e.*, rich transformation

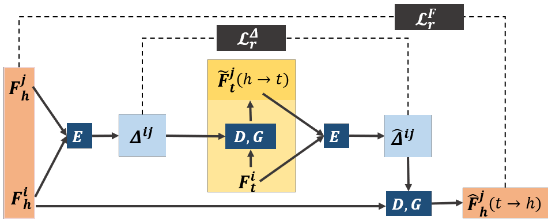


Fig. 4. The illustration of cycle delta reconstruction and feature reconstruction

information) between $\boldsymbol{F}_h^i$ and $\boldsymbol{F}_h^j$. By feeding various pairs of features from the same head class into delta feature extraction module E, we can obtain amounts of diverse delta features $\boldsymbol{\Delta}$, which can be applied to tail-class features to enlarge the feature space of tail class.

Sample-adaptive generation The sample-adaptive delta generator D and feature generator G are designed to produce sample-specific delta and features. The delta $\boldsymbol{\Delta}^{ij}$ extracted from different paired head-class features $\{\boldsymbol{F}_h^i, \boldsymbol{F}_h^j\}$ may vary due to complicated scene geometry and light sources, which may lead to different compatibility with different tail-class samples. Thus, it is crucial to attend relevant information from delta $\boldsymbol{\Delta}^{ij}$ according to tail-class feature $\boldsymbol{F}_t^i$ to produce sample-specific delta feature $\boldsymbol{\Delta}_t^{ij}$ more compatible to tail-class feature $\boldsymbol{F}_t^i$. D is designed to attend relevant variance information from extracted delta $\boldsymbol{\Delta}^{ij}$ according to specific tail-class feature $\boldsymbol{F}_t^i$ to produce sample-adaptive delta feature $\boldsymbol{\Delta}_t^{ij} \in \mathcal{R}^{C_{\boldsymbol{\Delta}} \times W \times H}$, where $\boldsymbol{\Delta}_t^{ij} = D\left(\text{concat}\left(\boldsymbol{\Delta}^{ij}, \boldsymbol{F}_t^i\right)\right)$. Then, we combine it with $\boldsymbol{F}_t^i$ into the generator G to produce augmented tail-class feature $\tilde{\boldsymbol{F}}_t^j$ belonging to class y_t^j:

$$\tilde{\boldsymbol{F}}_t^j = G(\boldsymbol{\Delta}_t^i + \boldsymbol{F}_t^i). \tag{3}$$

Cycle reconstruction loss To enforce delta extractor E to extract effective class-irrelevant delta feature and ensure the augmented features are faithful to input tail-class features (*i.e.*, to be sample-specific), we apply cycle reconstruction loss [19] in SAFA. We use objective functions that encourage reconstruction in *feature direction*: paired head-class feature $\{\boldsymbol{F}_h^i, \boldsymbol{F}_h^j\} \to \boldsymbol{\Delta}^{ij} \to$ reconstructed head-class feature $\hat{\boldsymbol{F}}_h^j$, and *delta direction*: $\boldsymbol{\Delta}^{ij} \to$ augmented tail-class feature $\tilde{\boldsymbol{F}}_t^j \to \hat{\boldsymbol{\Delta}}^{ij}$. For delta direction, with augmented paired tail-class features $\{\tilde{\boldsymbol{F}}_t^j, \boldsymbol{F}_t^i\}$, we can extract reconstructed class-irrelevant delta $\hat{\boldsymbol{\Delta}}^{ij} = E(\tilde{\boldsymbol{F}}_t^j - \boldsymbol{F}_t^i)$ and optimize:

$$\mathcal{L}_{\mathrm{r}}^{\boldsymbol{\Delta}} = ||\hat{\boldsymbol{\Delta}}^{ij} - \boldsymbol{\Delta}^{ij}||_2. \tag{4}$$

Note that $\hat{\boldsymbol{\Delta}}^{ij}$ and $\boldsymbol{\Delta}^{ij}$ are extracted from tail class and head class respectively, which means $\mathcal{L}_{\mathrm{r}}^{\boldsymbol{\Delta}}$ can force $\boldsymbol{\Delta}^{ij}$ to be class-irrelevant. For feature reconstruction direction, reconstructed delta feature $\hat{\boldsymbol{\Delta}}^{ij}$ combined with head-class feature $\boldsymbol{F}_h^i$ is fed into sample-adaptive generation module G to produce reconstructed head-class feature $\hat{\boldsymbol{F}}_h^j = G(D(\text{concat}(\hat{\boldsymbol{\Delta}}^{ij}, \boldsymbol{F}_h^i)) + \boldsymbol{F}_h^i)$, then we have:

$$\mathcal{L}_{\text{r}}^{F} = ||\hat{\boldsymbol{F}}_h^j - \boldsymbol{F}_h^j||_2. \tag{5}$$

The recycle reconstruction loss $\mathcal{L}_{\text{r}} = \mathcal{L}_{\text{r}}^{\boldsymbol{\Delta}} + \mathcal{L}_{\text{r}}^{F}$ can enforce delta extraction module E, sample-adaptive generation module D and G to work consistently for extracting transferable delta and adaptively combining delta with tail-class feature to produce sample-specific tail-class feature.

Contrastive loss To ensure the category-preserving characteristics of augmented tail-class features, we adopt a contrastive module Q and calculate the contrastive loss to ensure that the delta feature $\boldsymbol{\Delta}^{ij}$ not leaking head class information to augmented tail-class feature (*i.e.*, $\boldsymbol{\Delta}^{ij}$ is class-irrelevant). In a mini-batch $\boldsymbol{F}_a$ consisting of real head-class features $\boldsymbol{F}_h$, real tail-class features $\boldsymbol{F}_t$, and augmented tail-class features $\tilde{\boldsymbol{F}}_t$, we shuffle all samples with batch size s, and we form $s/2$ pairs by random sampling for training the contrastive module Q. Using $y_c \in \{0, 1\}$ as ground-truth to show whether the paired features come from the same class.

$$\mathcal{L}_c = -\left\langle \left(y_c \log \beta + (1 - y_c) \log (1 - \beta) \right) \right\rangle_{\frac{s}{2}} \tag{6}$$

where $\beta = Q(\boldsymbol{F}_a^i, \boldsymbol{F}_a^j)$ represent the probability distribution to show whether $\{\boldsymbol{F}_a^i, \boldsymbol{F}_a^j\}$ belong to the same class, and $\langle \cdot \rangle_{\frac{s}{2}}$ denotes that $\mathcal{L}_c$ is calculated over $s/2$ paired features on average.

Mode seeking loss To further produce diverse augmented tail-class features and enlarge feature space of tail classes, we employ mode seeking loss [29] to increase the distance between paired augmented tail-class features generated from the same $\boldsymbol{\Delta}^{ij}$ feature, and also extend the distance between a pair of augmented tail-class feature generated from the same tail-class feature, respectively. In detail, given delta feature $\boldsymbol{\Delta}^{ij}$ extracted from $\{\boldsymbol{F}_h^i, \boldsymbol{F}_h^j\}$ and paired features $\{\boldsymbol{F}_t^i, \boldsymbol{F}_t^j\}$ from the same tail class, we can produce paired augmented feature $\{\tilde{\boldsymbol{F}}_t^i, \tilde{\boldsymbol{F}}_t^j\}$ following Eqn. (3), the mode seeking loss can be written as:

$$\mathcal{L}_{\text{ms}}^{t} = \left\langle \frac{||\boldsymbol{F}_t^i - \boldsymbol{F}_t^j||_1}{||\tilde{\boldsymbol{F}}_t^i - \tilde{\boldsymbol{F}}_t^j||_1} \right\rangle_{\frac{s}{2}}, \mathcal{L}_{\text{ms}}^{h} = \left\langle \frac{||\boldsymbol{F}_h^i - \boldsymbol{F}_h^j||_1}{||\tilde{\boldsymbol{F}}_t^i - \tilde{\boldsymbol{F}}_t^j||_1} \right\rangle_{\frac{s}{2}}. \tag{7}$$

4 Experiment

We conduct experiment on CIFAR-LT-10/CIFAR-LT-100 [25], ImageNet-LT [27], Places-LT [47], and iNaturalist 2018 [37]. For those comparison experiments conducted in the same settings, we directly quote their results from original papers. Next, we briefly introduce these datasets and basic experiment settings. The details of datasets and implementation are reported in Supplementary.

4.1 Implementation Details and Datasets

In following experiments, our SAFA is employed before the second-to-last down-sampling layer, since we got the best results. In addition, we report additional experimental results on CIFAR-LT-10/CIFAR-LT-100 by integrating SAFA into different layers in Supplementary. The hyperparameter λ_1 (*resp.,* λ_2,λ_3,and λ_4) is set as 100 (*resp.,*, 1e-2, 1e2, 1e-1), and by observing validation accuracy on CIFAR-LT-100 dataset. We provide in-depth analysis of each loss item in Sec 4.3 and Supplementary.

During training, given a thresh epoch $\mathbb{T}_{th}$, which decides when to activate SAFA module to produce new tail-class features. Before $\mathbb{T}_{th}$, the network without SAFA is only optimized with $\mathcal{L}_{cls}$. After $\mathbb{T}_{th}$, SAFA is activated to be optimized with $\mathcal{L}$ in Eqn. (1) and produce augmented tail-class features. In each mini-batch, we sample same-class pairs from dataset, and they are split into two parts according to a manually set constant head-class ratio $\gamma = n_h/(n_h + n_t)$, where n_h and n_t denote the number of head classes and the number of tail classes, respectively. Following [38], we set head-class ratio $\gamma = 0.2$ for all datasets.

CIFAR-LT: For CIFAR-LT-10 (*resp.,* CIFAR-LT-100) with 10 (*resp.,* 100) classes, following [11], we create 5 training sets by changing the imbalance factor ρ in the range of $\{200, 100, 50, 20, 10\}$, where ρ is the image amount ratio between the largest and smallest classes. We use the original balanced test sets for our test sets. Following [38], the main results on CIFAR-LT-10/CIFAR-LT-100 are trained on ResNet-32 [16] for 200 epochs with batch size of 128. The learning rate was set to 0.1 at the beginning, then declined by 0.01 at the 160-th epoch and again at the 180-th epoch. Our SAFA is activated at $\mathbb{T}_{th} = 159$.

ImageNet-LT: ImageNet-LT is built in [27] based on ImageNet dataset [32] with 1000 classes, its imbalance factor ρ is 1280/5. Our experiments about ImageNet-LT are conducted with ResNeXt-50-32x4d [16], which was trained with a batch size of 256 for 100 epochs, as described in [22]. The initial learning rate was set to 0.1, and it gradually declined by 0.1 at the 60-th, 80-th, and 95-th epochs, respectively. According to [38], test set classes are further divided into three groups: many-shot (over 100 samples), medium-shot (between 20 and 100 samples), and few-shot (less than 20 samples) to better examine performance differences across classes with different numbers of samples seen during training. Our SAFA is integrated into ResNeXt-50-32x4d at $\mathbb{T}_{th} = 59$.

Places-LT: Places-LT is a subset of the large-scale scene classification dataset [47]. The dataset comprises 365 categories with class cardinality ranging from 5 to 4980. Following [27], we finetune ResNet-152, which is pre-trained on the entire ImageNet dataset [32]. The network was trained with a batch size of 256. The starting learning rate was set to 0.01, and it declined by 0.1 every ten epochs until the training was terminated after 30 epochs. Our SAFA is employed at $\mathbb{T}_{th} = 9$. Similar to the ImageNet-LT evaluation, the top-1 accuracy of many-shot, medium-shot, and few-shot in this study are reported.

iNaturalist 2018: The iNaturalist 2018 [37] dataset is a large-scale dataset with images collected from 8142 classes in real-world, which have an extremely imbalanced class distribution with an imbalance factor of 1000/2. With a batch size of 256, we train ResNet-50 from scratch across 90 epochs. The learning rate was initially set to 0.1 and then degraded by 0.1 at the 50-th, 70-th, and 85-th

Dataset	CIFAR-10					CIFAR-100				
Imbalance factor	200	100	50	20	10	200	100	50	20	10
CE loss	34.13	29.86	25.06	17.56	13.82	64.44	61.23	55.21	48.06	42.43
CB-CE[11]	31.23	27.32	21.87	15.44	13.10	64.44	61.23	55.21	48.06	42.43
CB fine-tuning[12]	33.76	28.66	22.56	16.78	16.83	61.34	58.50	53.78	47.70	42.43
L2RW[31]	33.75	27.77	23.55	18.65	17.88	67.00	61.10	56.83	49.25	47.88
Meta-weight[35]	32.80	26.43	20.90	15.55	12.45	63.38	58.39	54.34	46.96	41.09
CB-RSG [38]	30.96	25.68	20.25	15.26	12.24	62.69	57.94	54.40	46.23	42.31
CB-SAFA	**27.18**	**23.68**	**19.79**	**14.01**	**12.07**	**60.34**	**54.13**	**52.04**	**44.56**	**39.77**
Focal loss [26]	34.71	29.62	23.29	17.24	13.34	64.38	61.59	55.68	48.05	44.22
CB Focal loss[11]	31.85	25.43	20.78	16.22	12.52	63.77	60.40	54.79	47.41	42.01
Focal loss-RSG[38]	30.12	26.11	21.58	14.98	12.51	62.81	57.61	54.85	46.31	42.53
Focal loss-SAFA	**25.68**	**21.58**	**18.58**	**13.96**	**12.21**	**61.52**	**54.32**	**52.23**	**44.08**	**40.58**
LDAM loss[6]	33.25	26.45	21.17	16.11	12.68	63.47	59.40	53.84	48.41	42.71
LDAM-DRW[6]	25.26	21.88	18.73	15.10	11.63	61.55	57.11	52.03	47.01	41.22
LDAM-DRW-RSG[38]	26.04	21.74	17.32	13.71	11.55	60.85	55.45	51.50	45.76	42.03
LDAM-DRW-SAFA	**22.47**	**19.52**	**16.43**	**13.62**	**11.06**	**57.53**	**53.96**	**49.98**	**44.12**	**40.89**

Table 1. Test top-1 errors (%) of ResNet-32 on CIFAR-LT-10 and CIFAR-LT-100 with imbalance ratio ρ ranging from $\{200, 100, 50, 20, 10\}$

epochs, respectively. Our SAFA is utilized at $\mathbb{T}_{th} = 69$, and we report top-1 err as final evaluation.

4.2 Comparison with Previous Methods

Considering that our SAFA worked as a plug-in can be integrated different networks and combined with different loss functions, here, we conduct comparison experiment on typical long-tailed methods [11,26,6] and several state-of-the-art methods [38,45,20]. For the sake of brevity, we will refer to the baseline trained using cross-entropy (*resp.,* Class-Balanced Cross-Entropy losss [11]) as "CE loss" (*resp.,* "CB-CE loss"), and refer to "A-SAFA" as a combination of our SAFA and the method "A" .

Results on CIFAR-LT: Comparison result on CIFAR-LT-10 and CIFAR-LT-100 with imbalance factor ρ ranging from $\{200, 100, 50, 20, 10\}$ are shown in Table 1, which are categorised into three groups according to the adopted basic losses (*i.e.,* CE, focal [26], and LDAM [6]). We evaluate our method with the three basic losses. The results reveal that our method can consistently improve the performance of the basic losses significantly. Particularly, our method notably surpasses mixup that conducts augmentation on the inputs and RSG [38] that augments tail class by leveraging knowledge from tail class, manifesting that our augmentation method is more effective in long-tailed scenarios. Furthermore, SAFA outperforms the re-weighting strategies. This illustrates that our augmentation method can indeed improve classifier performance. SAFA can still obtain stable performance gains when the dataset is less imbalanced (implying imbalance factor ρ=10), demonstrating that SAFA will not harm the

Method	Many	Medium	Few	All
CE loss	65.9	37.5	7.7	44.4
Focal Loss [26]	63.3	37.4	7.7	43.2
OLTR [27]	52.1	39.7	20.3	41.2
Joint [22]	**65.9**	37.5	7.7	44.4
NCM [22]	56.6	45.3	28.1	47.3
cRT [22]	61.8	46.2	27.4	49.6
τ-normalized [22]	59.1	46.9	30.7	49.4
LWS [22]	60.2	47.2	30.3	49.9
LDAM-DRS [6]	63.7	47.6	30.0	51.4
LDAM-DRS-RSG [38]	63.2	48.2	32.3	51.8
LDAM-DRS-SAFA (ours)	63.8	**49.9**	**33.4**	**53.1**

Table 2. Top-1 accuracy of ResNeXt-50 on ImageNet-LT

Method	Many	Medium	Few	All
Lifted Loss [30]	41.1	35.4	24.0	35.2
Focal Loss [26]	41.1	34.8	22.4	34.6
Range Loss [46]	41.1	35.4	23.2	35.1
FSLwF [13]	43.9	29.9	29.5	34.9
BBN[48]	42.5	40.3	30.6	38.7
OLTR [27]	44.7	37.0	25.3	35.9
τ-normalized [22]	37.8	40.7	31.8	37.9
LDAM-DRS [6]	43.3	38.3	30.7	38.6
DisAlign [45]	40.4	42.4	30.1	39.3
LDAM-DRS-RSG [38]	41.9	41.4	32.0	39.3
LDAM-DRS-SAFA (Ours)	42.1	**42.7**	**33.4**	**41.5**

Table 3. Top-1 accuracy of ResNet-152 on Places-LT.

Method	Error Rate
CB Focal Loss[11]	38.88
CE-DRW [6]	36.27
CE-DRS [6]	36.44
BBN [48]	33.71
τ-normalized [22]	34.40
LDAM-DRW [6]	34.00
LDAM-DRS [6]	32.73
LDAM-DRW-SSP [20]	33.70
DisAlign [45]	32.20
LDAM-DRW-RSG [38]	33.22
LDAM-DRS-RSG [38]	32.10
LDAM-DRS-SAFA (Ours)	**30.22**

Table 4. Top-1 error rates of ResNet-50 on iNaturalist 2018

classifier's performance in a moderately balanced scenario. Another observation is that re-weighting strategies [11,6] are beneficial for long-tailed issues, since some re-weighting methods including CB-CE, CB Focal loss, CB-RSG, as well as our CB-SAFA surpass cross-entropy training (CE loss) by a significant margin. Moreover, we compare our method with other previous sample generation methods [33,43,39] in Supplementary.

Results on ImageNet-LT: We present the results for ImageNet-LT in Table 2. When compared to LDAM-DRS, LDAM-DRW-RSG [38], LADM-DRS-SAFA (ours) still achieves a greater level of accuracy, demonstrating that SAFA can solve the problem of imbalanced datasets. On medium-shot and few-shot classes, SAFA can produce effective and diverse tail-class features to enlarge tail-class feature space to improve the model and considerably improve its generality.

Results on Places-LT: The Table 3 shows the top-1 accuracy on Place-LT. The results reveal that when SAFA is paired with LDAM-DRS, performance may be increased even further, demonstrating that SAFA is useful. Furthermore, when compared to the two most current prominent approaches, tau-normalized, BBN, DisAlign, and RSG, SAFA can increase the model's performance on medium-

Dataset	CIFAR-10				CIFAR-100			
Networks	$\rho = 200$		$\rho = 10$		$\rho = 200$		$\rho = 10$	
	w/o SAFA	SAFA	w/o SAFA	SAFA	w/o SAFA	SAFA	w/o SAFA	SAFA
ResNet-32	25.26	**23.42**	11.63	**11.06**	61.55	**58.31**	41.22	**41.02**
ResNet-56	23.59	**21.49**	10.35	**10.12**	59.71	**56.37**	39.69	**39.21**
ResNet-110	23.18	**21.09**	10.04	**9.86**	59.13	**55.17**	39.07	**38.51**
DenseNet-40	22.92	**20.56**	9.94	**9.56**	58.96	**54.87**	38.81	**38.17**
ResNeXt-29	22.81	**20.44**	9.71	**9.39**	58.97	**54.79**	38.74	**38.15**

Table 5. Top-1 error rates of different network architectures combined with LDAM-DRW [6] on CIFAR-LT

Dataset	CIFAR-10					CIFAR-100				
Imbalance factor	200	100	50	20	10	200	100	50	20	10
LDAM-DRW[6]	25.26	21.88	18.73	15.10	11.63	61.55	57.11	52.03	47.01	41.22
w/o $\mathcal{L}_r$	31.21	26.96	20.97	18.74	13.15	64.87	62.41	56.83	49.12	42.53
w/o $\mathcal{L}_{ms}^{\boldsymbol{t}}$	23.56	20.39	17.61	14.83	13.18	58.87	54.38	51.32	45.92	42.87
w/o $\mathcal{L}_{ms}^{\boldsymbol{h}}$	24.67	20.94	17.16	13.67	11.69	59.89	55.09	51.29	44.86	41.78
w/o $\mathcal{L}_c$	23.84	20.41	17.08	13.89	11.54	58.53	54.13	50.71	44.69	41.53
Full method	**22.47**	**19.52**	**16.43**	**13.62**	**11.06**	**57.53**	**53.96**	**49.98**	**44.12**	**40.89**

Table 6. Results of ablated methods by removing each proposed loss from Eqn (1). We report the top-1 error rates of ResNet-32 combined with SAFA and LDAM-DRW [6] on CIFAR-LT-10/CIFAR-LT-100 with different imbalance ratios

shot and few-shot classes while causing less accuracy loss on many-shot classes, resulting in higher overall accuracy and competitive result.

Results on iNaturalist 2018: We show the experimental results under the same setting as [38] on iNaturalist 2018 dataset. The results reveal that by leveraging the proposed sample-adaptive feature augmentation method, we may achieve superior results, demonstrating the efficacy of SAFA. As can be observed, SAFA assists the model in achieving competitive outcomes, demonstrating that SAFA is capable of effectively coping with imbalanced datasets.

4.3 Ablation Studies

Adaptivity to different backbone networks: Firstly, we analyze the effectiveness of our proposed SAFA module by integrating SAFA into different network architectures including ResNet-32, ResNet-56, ResNet-110, DenseNet-40, and ResNeXt-29 (8×64d), and report the comparison results on CIFAR-LT-10/CIFAR-LT-100 with $\rho = \{200, 10\}$ in Table 5, in which “w/o SAFA” denotes that removing SAFA during training. From Table 5, we can see that all models equipped with SAFA are consistently better whether $\rho = 100$ or $\rho = 10$, which indicates that SAFA can be employed into various deep neural network to improve long-tail classification performance.

Combination with different loss functions: In table 1, by comparing the results of CE loss (*resp.,* Focal loss, LDAM-DRW loss) with the results of CB-SAFA (*resp.,* SAFA focal loss, LDAM-DRW SAFA loss), it is seen that our SAFA is compatible with different loss functions and can consistently improve classification performance based on different loss functions.

Analysis of each loss term of SAFA: In our SAFA, we employ a reconstruction loss $\mathcal{L}_r$, a tail mode seeking loss $\mathcal{L}^t_{ms}$, a head mode seeking loss $\mathcal{L}^h_{ms}$, and a contrastive loss $\mathcal{L}_c$. To investigate the impact of each loss term, we conduct ablation studies on CIFAR-10 and CIFAR-100 datasets by removing each loss term from the final objective in Eqn. (1). The results are summarized in Table 6. Firstly, we can see that the classification performance is compromised when removing $\mathcal{L}_r$, even worse than baseline LDAM-DRW [6] without augmentation, implying that our recycle reconstruction loss is necessary and it enforces our SAFA module to extract transferable delta and achieve sample-adaptive augmentation. Removing $\mathcal{L}_c$ results in slight performance degradation on two datasets, since the generated features may not belong to the category of combined tail class without contrastive loss. By removing the head mode seeking loss $\mathcal{L}^h_{ms}$, we can see that the classification performance in less imbalanced scenarios such as imbalance ratio $\rho = \{200, 100\}$ on two datasets become much worse while leaving less impact on relatively balanced settings with $\rho = \{20, 10\}$. Another observation is that ablating tail mode seeking loss $\mathcal{L}^t_{ms}$ results in a minor deterioration of classification performance in extremely imbalanced settings with $\rho = \{200, 100\}$, compared to a more significant decline in less imbalanced settings with $\rho = \{20, 10\}$. It can be explained as follows: in extremely imbalanced settings like $\rho = \{200, 100\}$, where head-class samples may be compact in feature space, leading to more compact deltas in feature space, in other words, the distance among deltas is limited. In this scenario, using head mode seeking loss $\mathcal{L}^h_{ms}$ to enlarge the distance between real tail-class feature and augmented tail-class feature based on the distance of head-class pairs can produce diverse tail-class samples, whereas the distance between deltas may be sufficient to produce different samples without the use of $\mathcal{L}^h_{ms}$. Similarly, $\mathcal{L}^t_{ms}$ is adopted to enlarge the distance between two tail-class features augmented from the same tail-class feature. It is helpful to leverage $\mathcal{L}^t_{ms}$ to enforce SAFA to be sensitive to the difference of paired features from the same tail class in a less imbalanced setting, where the tail-class feature space may be compact, however it is not necessary for a relatively loose tail-class feature space.

5 Conclusions

In this paper, we propose a novel plug-in approach SAFA, which is convenient to be integrated into various networks and coupled with different loss functions. Our SAFA aims to extract transferable delta from head class and achieve sample-adaptive application to tail class to enlarge tail-class feature space. Extensive experiment demonstrate the effectiveness of the proposed SAFA.

References

1. Almahairi, A., Rajeswar, S., Sordoni, A., Bachman, P., Courville, A.C.: Augmented cyclegan: Learning many-to-many mappings from unpaired data. In: ICML (2018)
2. Antoniou, A., Storkey, A., Edwards, H.: Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340 (2017)
3. Bengio, Y.: Learning deep architectures for AI (2009)
4. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks **106**, 249–259 (2018)
5. Byrd, J., Lipton, Z.: What is the effect of importance weighting in deep learning? In: ICML (2019)
6. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. NeurIPS (2019)
7. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
8. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In: CVPR (2018)
9. Chou, H.P., Chang, S.C., Pan, J.Y., Wei, W., Juan, D.C.: Remix: rebalanced mixup. In: ECCV (2020)
10. Chu, P., Bian, X., Liu, S., Ling, H.: Feature space augmentation for long-tailed data. In: ECCV (2020)
11. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: CVPR (2019)
12. Cui, Y., Song, Y., Sun, C., Howard, A., Belongie, S.: Large scale fine-grained categorization and domain-specific transfer learning. In: CVPR (2018)
13. Gidaris, S., Komodakis, N.: Dynamic few-shot visual learning without forgetting. In: CVPR (2018)
14. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014)
15. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE TKDE **21**(9), 1263–1284 (2009)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
17. Huang, C., Li, Y., Loy, C.C., Tang, X.: Learning deep representation for imbalanced classification. In: CVPR (2016)
18. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017)
19. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: (ECCV) (2018)
20. Jamal, M.A., Brown, M., Yang, M.H., Wang, L., Gong, B.: Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In: CVPR (2020)
21. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intelligent data analysis **6**(5), 429–449 (2002)
22. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. In: ICLR (2019)
23. Kim, J., Jeong, J., Shin, J.: M2m: Imbalanced classification via major-to-minor translation. In: CVPR (2020)

24. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
25. Krizhevsky, A., et al.: Learning multiple layers of features from tiny images (2009)
26. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017)
27. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: CVPR (2019)
28. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
29. Mao, Q., Lee, H.Y., Tseng, H.Y., Ma, S., Yang, M.H.: Mode seeking generative adversarial networks for diverse image synthesis. In: CVPR (2019)
30. Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: CVPR (2016)
31. Ren, M., Zeng, W., Yang, B., Urtasun, R.: Learning to reweight examples for robust deep learning. In: ICML (2018)
32. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet: large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)
33. Schwartz, E., Karlinsky, L., Shtok, J., Harary, S., Marder, M., Kumar, A., Feris, R., Giryes, R., Bronstein, A.: Delta-encoder: an effective sample synthesis method for few-shot object recognition. NeurIPS (2018)
34. Shen, L., Lin, Z., Huang, Q.: Relay backpropagation for effective learning of deep convolutional neural networks. In: ECCV (2016)
35. Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., Meng, D.: Meta-weight-net: Learning an explicit mapping for sample weighting. NeurIPS (2019)
36. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
37. Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., Belongie, S.: The inaturalist species classification and detection dataset. In: CVPR (2018)
38. Wang, J., Lukasiewicz, T., Hu, X., Cai, J., Xu, Z.: Rsg: A simple but effective module for learning imbalanced datasets. In: CVPR (2021)
39. Wang, Y.X., Girshick, R., Hebert, M., Hariharan, B.: Low-shot learning from imaginary data. In: CVPR (2018)
40. Wang, Y.X., Ramanan, D., Hebert, M.: Learning to model the tail. NeurIPS (2017)
41. Wang, Y., Pan, X., Song, S., Zhang, H., Huang, G., Wu, C.: Implicit semantic data augmentation for deep networks. NeurIPS (2019)
42. Wu, T., Huang, Q., Liu, Z., Wang, Y., Lin, D.: Distribution-balanced loss for multi-label classification in long-tailed datasets. In: ECCV (2020)
43. Yin, X., Yu, X., Sohn, K., Liu, X., Chandraker, M.: Feature transfer learning for deep face recognition with under-represented data. arXiv preprint arXiv:1803.09014 (2018)
44. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: ICLR (2018)
45. Zhang, S., Li, Z., Yan, S., He, X., Sun, J.: Distribution alignment: A unified framework for long-tail visual recognition. In: CVPR (2021)
46. Zhang, X., Fang, Z., Wen, Y., Li, Z., Qiao, Y.: Range loss for deep face recognition with long-tailed training data. In: ICCV (2017)
47. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. PAMI **40**(6), 1452–1464 (2017)
48. Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.: Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: CVPR (2020)

49. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017)